CLAIMS:

What is claimed is:

1. In a data processing system having a coherent memory hierarchy that includes a memory and a plurality of caches each assigned to particular ones of a plurality of devices that generate cache access operations, a method of maintaining cache coherency comprising:

a first device issuing an address operation requesting sole ownership of a cache line that said first device intends to overwrite in a first cache;

changing a coherency state of the cache line within said first cache to a first coherency state that indicates that the first device has sole ownership of the cache line and may or may not overwrite the cache line;

in response to snooping said address operation, changing a coherency state of the cache line in a second cache associated with a snooping device to a second state without sending data from said cache line in the second cache to the first cache;

wherein sole ownership of said cache line is provided to said first device without data being sourced to said first cache from another cache.

2. The method of Claim 1, wherein, when said first device subsequently initiates a write of said cache line, said method further comprises changing said first state to a third state indicating that a most coherent copy of said data exists within the cache line of the first cache.

3. The method of Claim 1, wherein said address operation for sole ownership of a cache line is generated for a speculatively issued cache line overwrite operation, said method further comprising:

speculatively issuing the address operation for sole ownership of the cache line; and

determining whether said cache line overwrite operation was correctly speculated, wherein said first coherency state is changed to another coherency state depending on whether said cache line overwrite operation was correctly speculated.

4. The method of Claim 3, wherein when said cache line overwrite operation was not correctly speculated, said method further comprises:

changing the coherency state of the cache line in the first cache from said first coherency state to an invalid state; and

subsequently sourcing requests for said cache line from memory.

5. The method of Claim 3, wherein when said cache line overwrite operation was correctly speculated, said method further comprises:

initiating a write of said cache line with data provided by said first device; and

changing said first state to a third state indicating that a most coherent copy of said data exists within the cache line of the first cache.

6. The method of Claim 2, further comprising:

snooping requests for access to said cache line at said first cache;

when the cache line in said first cache is in the third coherency state and said first device has completed writing data to said first cache, sourcing the data from the first cache to the second cache;

when the cache line in said first cache is in the invalid coherency state, sourcing the data from memory.

7. The method of Claim 2, further comprising:

snooping requests for access to said cache line at said first cache; and

when the cache line in the first cache is still in the first coherency state, retrying all snooped requests, wherein all subsequent requests snooped while said cache line is in the first coherency state is retried until the coherency state changes.

8. The method of Claim 2, further comprising:

snooping a read requests for said cache line at said first cache; and

when the read request receives a null response and said cache line in the first cache is still in the first coherency state:

sourcing data for the cache line from memory; and

changing said first coherency state to an invalid state in said first cache.

9. The method of Claim 1, wherein said first device is an I/O device and said first cache is an I/O cache controlled by an I/O controller, said method further comprising:

issuing the address operation as a direct memory access (DMA) Claim in response to a speculative DMA write.

10. The method of Claim 1, wherein said first device is a processor and said first cache is a processor cache controlled by an cache controller, said method further comprising:

issuing the address operation in response to a data cache block zero (DCBZ) operation.

11. In a data processing system having a memory hierarchy that includes a memory and a plurality of caches interconnected by a system bus and each accessible by particular ones of a plurality of devices, a caching mechanism that provides address coherency operations for cache line writes by a first device, said caching mechanism comprising:

a first cache line that has a corresponding cache line in a second cache associated with a second device;

a coherency tracking mechanism that supports at least a first coherency state, a second coherency state and a third coherency state, wherein:

said first coherency state indicates that data within said first cache line is currently invalid but may or may not be overwritten by said first device;

said second coherency state indicates that the data is invalid; and

said third coherency state indicates that the data is a most coherent copy of said data;

means for a first device to issue an address operation requesting sole access to said cache line that said first device intends to overwrite in a first cache;

means for changing a coherency state of said cache line within said first cache to said first coherency state when a response is received on said system bus indicating that sole ownership has been granted to said first cache.

12. The caching mechanism of Claim 11, further comprising:

means for snooping the address operation; and

means, when the address operation is snooped while the cache line is in said third coherency state within the snooping device's cache and the snooped operation is for access that does not overwrite the entire cache line, for:

issuing data from the snooping device's cache line on the data bus when said access is granted to said first device; and

changing a coherency state of the snooping device's cache line to another state that indicates that the first device's cache line has data in a coherency state that is as coherent or more coherent than said snooping device's cache line; and

means, when the snooped request is received while the snooping device's cache line is in said third coherency state and the snooped operation is for sole ownership of the cache line that is to be completely overwritten by the first device, for changing a coherency state of the snooping device's cache line to said second state, and withholding any transfer of data to the first device's cache line.

13. The caching mechanism of Claim 12, further comprising:

means for overwriting the data within said first device's cache line with data from the first device, wherein said overwriting is only initiated after sole ownership has been granted and said first cache line is in said first coherency state; and

subsequently changing the coherency state of the first device's cache line to the third state indicating that a most coherent copy of said data exists within the first device's cache line.

14. The caching mechanism of Claim 13, further comprising:

means for snooping requests for access to said cache line from a requesting device;

means, when the cache line is in the third coherency state and said first device has completed writing data to said cache line, for sourcing the data from the cache line to the requesting device;

means, when the cache line is in the second coherency state, for indicating that said data should be sourced from memory.

15. The caching mechanism of Claim 13, further comprising:

means for snooping requests for access to said cache line from a requesting device; and

means, when the first device's cache line is still in the first coherency state, for retrying all snooped requests, wherein all subsequent requests snooped while said first device's cache line is in the first coherency state are retried until the coherency state changes to a different coherency state.

16. The caching mechanism of Claim 13, further comprising:

means for snooping a read requests for said cache line; and

means, when the read request receives a null response and said cache line is still in the first coherency state, for:

indicating that data for the cache line should be sourced from memory; and

changing said first coherency state to the second coherency state.

17. The caching mechanism of Claim 11, wherein said address operation for sole ownership of a cache line is generated for a speculatively issued cache line overwrite operation, said system further comprising:

means for speculatively issuing the address operation for sole ownership of the cache line by the first device; and

means for determining whether said cache line overwrite operation was correctly speculated, wherein said first coherency state is changed to another coherency state depending on whether said cache line overwrite operation was correctly speculated.

18. The caching mechanism of Claim 17, wherein when said cache line overwrite operation was not correctly speculated, said mechanism further comprises:

means for changing the coherency state of the first device's cache line in the first cache from said first coherency state to the second coherency state; and

means for subsequently sourcing requests for said cache line from memory.

19. The caching mechanism of Claim 17, wherein when said cache line overwrite operation was correctly speculated, said mechanism further comprises:

means for initiating a write of said first device's cache line with data provided by said first device; and

means for changing said first coherency state to the third coherency state in said first device's cache line.

20. The caching mechanism of Claim 1, wherein said first device is an I/O device and said caching mechanism includes an I/O cache controlled by an I/O controller, and further comprises:

means for issuing the address operation as a direct memory access (DMA) Claim in response to a speculative DMA write.

21. The caching mechanism of Claim 1, wherein said first device is a processor and said caching mechanism includes a processor cache controlled by a cache controller, and further comprises means for issuing the address operation in response to a data cache block zero (DCBZ) operation.

22. A data processing system, comprising:

an interconnect including an address bus and a data bus;

a plurality of devices interconnected via coupled to said interconnect;

a plurality of caches that are each associated with a device among said plurality of devices, wherein a first cache associated with a first device includes:

a cache line of a cache having a coherency indicator and coherency mechanism that supports at least a first coherency state, a second coherency state and a third coherency state, wherein:

said first coherency state indicates that data within said first cache line is currently invalid but may or may not be overwritten by said first device;

said second coherency state indicates that the data is invalid; and

said third coherency state indicates that the data is a most coherency copy of the data across the plurality of caches;

means for said first device to issue an address operation requesting sole access to said cache line that the first device intends to overwrite in the first cache;

means for changing a coherency state of said first device's cache line to said first coherency state when a response is received indicating that sole ownership has been granted to said first device.

23. The data processing system of Claim 22, further comprising:

means for a snooping device to snoop the address operation; and

means, when the snooping device snoops the address operation while the snooping device's cache line is in said third coherency state and the snooped operation is for access that does not overwrite the entire cache line, for:

issuing data from the snooping device's cache line on the data bus when said access is granted to said first device; and

changing a coherency state of the snooping device's cache line to another state that indicates that the first device's cache line has data in a coherency state that is as coherent or more coherent than said snooping device's cache line;

means, when the snooped operation is received while the snooping device's cache line is in said third coherency state and the snooped operation is for sole ownership of the cache line that is to be completely overwritten by the first device, for changing a coherency state of the snooping device's cache line to said second state, and withholding any transfer of data to the first device's cache line; and

means, when the snooped operation is received, for changing the coherency state of the cache line in the first cache to the first coherency state when sole access is granted to the first device's cache.